Magrathea Engine

Document :	**Specification of the realization**
Version :	1.0
Group leader :	Serge Lages
Group members :	David Callu Louise Jacquot Laurent Marchal Nicolas Remia Fabien Rioli
Date :	08/01/2006
Recipient :	Flavien Astraud

Abstract

The aim of this documentation is to give a technical description of all features provided by the Magrathea Engine and its designer. And it covers aspects of this project's organization and describe his goals. The Magrathea Engine is an environmental engine designed with powerful tools to provide a complete solution for managing realistic simulations.

© Copyright 2005 – PFE EPITECH

pfe@epitech.net

Table of content

1. PFE Reminder..... 3
1.1 – Subject..... 3
1.2 – Objectives of PFE..... 3
1.3 – Base principals of future system..... 3

2. Presentation of the realization's environment..... 5
2.1 – Realization's environment..... 5
2.2 – Hardware's environment..... 5
2.3 – Technical architecture..... 5
2.4 – Existing components used for the project..... 5
2.5 – Careful points..... 6

3. Programs description..... 7
3.1 – General architecture..... 7
3.2 – Technical description..... 8
3.3 – In/Out data..... 11

4. Annex..... 14
4.1 – General code's norm..... 14
4.2 – osgDesigner Use Case..... 15
4.3 – osgDesigner code norm..... 15

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

1. PFE Reminder

1.1 – Subject

We want to develop a new solution of environmental engine, based on the open source philosophy to provide an engine and tools allowing to create a real alternative. It will be based around the OpenSceneGraph project, an open source, cross platform graphics toolkit for the development of high performance graphics applications.

1.2 – Objectives of PFE

The stakes of this project are multiples :

- First, we want to help the open source community by improving the OpenSceneGraph project with a environmental engine and powerful tools.
- We want to fill a current leak in open source engines by developing a fast, reliable and easy-to-use solution for realistic simulations.

1.3 – Base principals of future system

This project will be composed of two main parts, an environmental engine Magrathea and a graphical tool osgDesigner used to edit 3D scenes. Here is the awaited results for each of them :

Engine objectives :

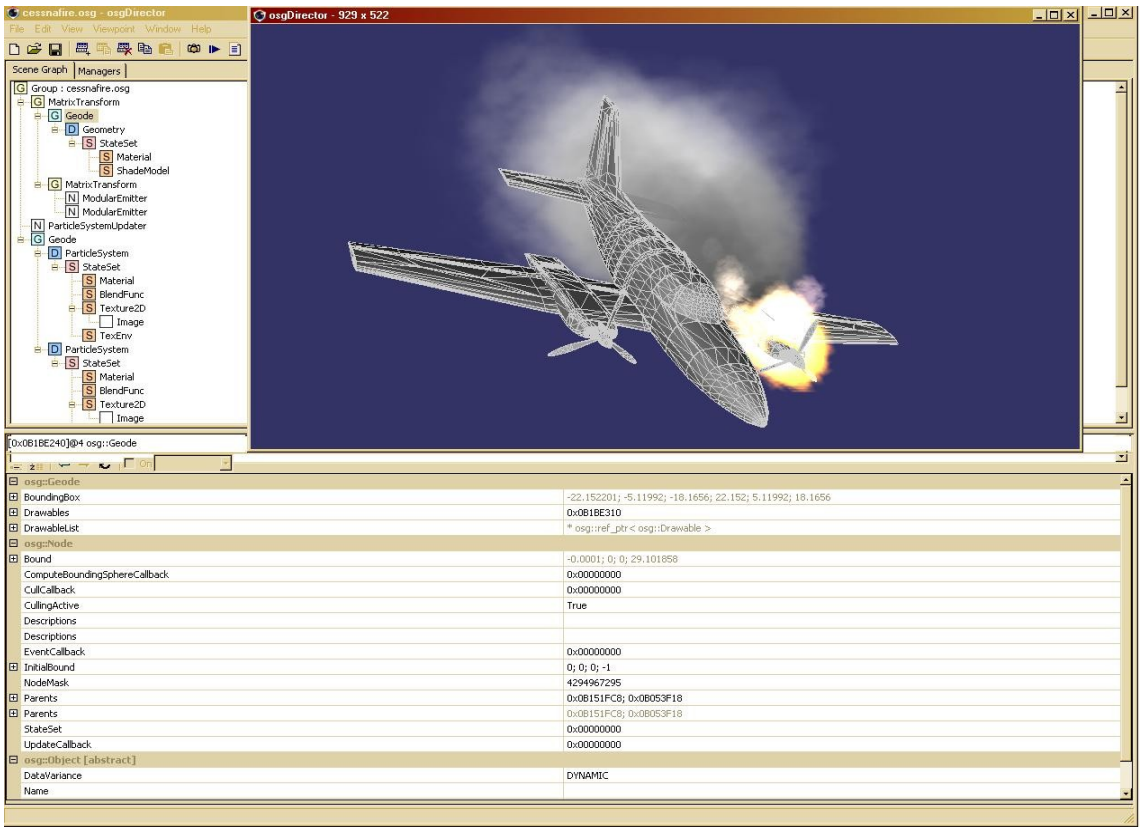
- Rendering realistic water comportments.
- Managing atmospheres and weather effects.
- Providing terrains without size limits, and with high zoom possibilities.
- Rendering realistic vegetation.



pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

Tools objectives :

- Provide an universal platform for editing OpenSceneGraph elements.
- Provide a very flexible plug in system: plugin developpers will be able to improve and extend osgDesigner at need.
- Develop a minimal set of plug ins for editing each kind of OpenSceneGraph objects and objects exported by openSceneGraph-base library.
- Provide a user-friendly interface, which looks like :



pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

2. Presentation of the realization's environment

2.1 – Realization's environment

We are developing on two different operating systems, with there own applications :

- On Windows : Visual Studio 2005 for editing and compiling our work.
- On UNIX : GNU/Emacs for editing and gcc for compilation.
- On All OS : Eclipse for editing and compiling our work.

For the source's management, we are using Tigris' SVN, a version control system that is a compelling replacement for CVS in the open source community.

2.2 – Hardware's environment

We are limited by the hardware material found in EPITA buildings:

- Pentium 4 3Ghz
- 1Go RAM.
- Integrated Intel graphic chipset.
- Rackable slots.

2.3 – Technical architecture

- This project is cross platform: we have to provide both engine and tools compatible with Windows and All UNIX-like OSes.
- The engine and the tools are written in standard C/C++.
- Both the tools and the engine will use the XML language.
- The Graphical User Interface used in the tool is Qt (v4)

2.4 – Existing components used for the project

We are going to use :

- OpenSceneGraph : It is an open source high performance 3D graphics toolkit, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modeling. Written entirely in Standard C++ and OpenGL it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris and FreeBSD operating systems.
- Producer : Open Producer (or simply Producer) is a cross-platform C++/OpenGL library that is focused on Camera control. Producer's Camera provides projection, field of view, viewpoint control, and frame control. Further, Producer can be used in a multi-tasking environment to allow multiple Camera's to run in parallel supporting hardware configurations with multiple display subsystems. Threading, Camera synchronization and frame rate control are simplified in the Producer programming interface.

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

- OpenThreads : This library is intended to provide a minimal & complete Object-Oriented (OO) thread interface for C++ programmers. It is loosely modeled on the Java thread API, and the POSIX Threads standards. The architecture of the library is designed around "swappable" thread models which are defined at compile-time in a shared object library.

- Expat : Expat is an XML parser library written in C. It is a stream-oriented parser in which an application registers handlers for things the parser might find in the XML document (like start tags).

2.5 – Careful points

The Magrathea engine must be :

- Performant
- Stable

And the tools must be :

- User-friendly
- Powerful

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

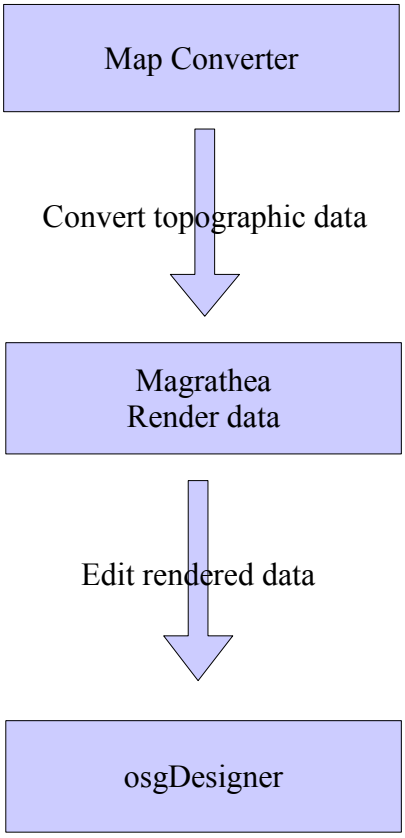
3. Programs description

3.1 – General architecture

The project is composed of :

- osgDesigner : an universal editor for OpenSceneGraph that will allow to edit specific parts of the engine.
- Map Converter : a tool to convert topographic data to the engine's file format.
- Magrathea : the environmental engine based on OpenSceneGraph.

Here is the links between theses programs :

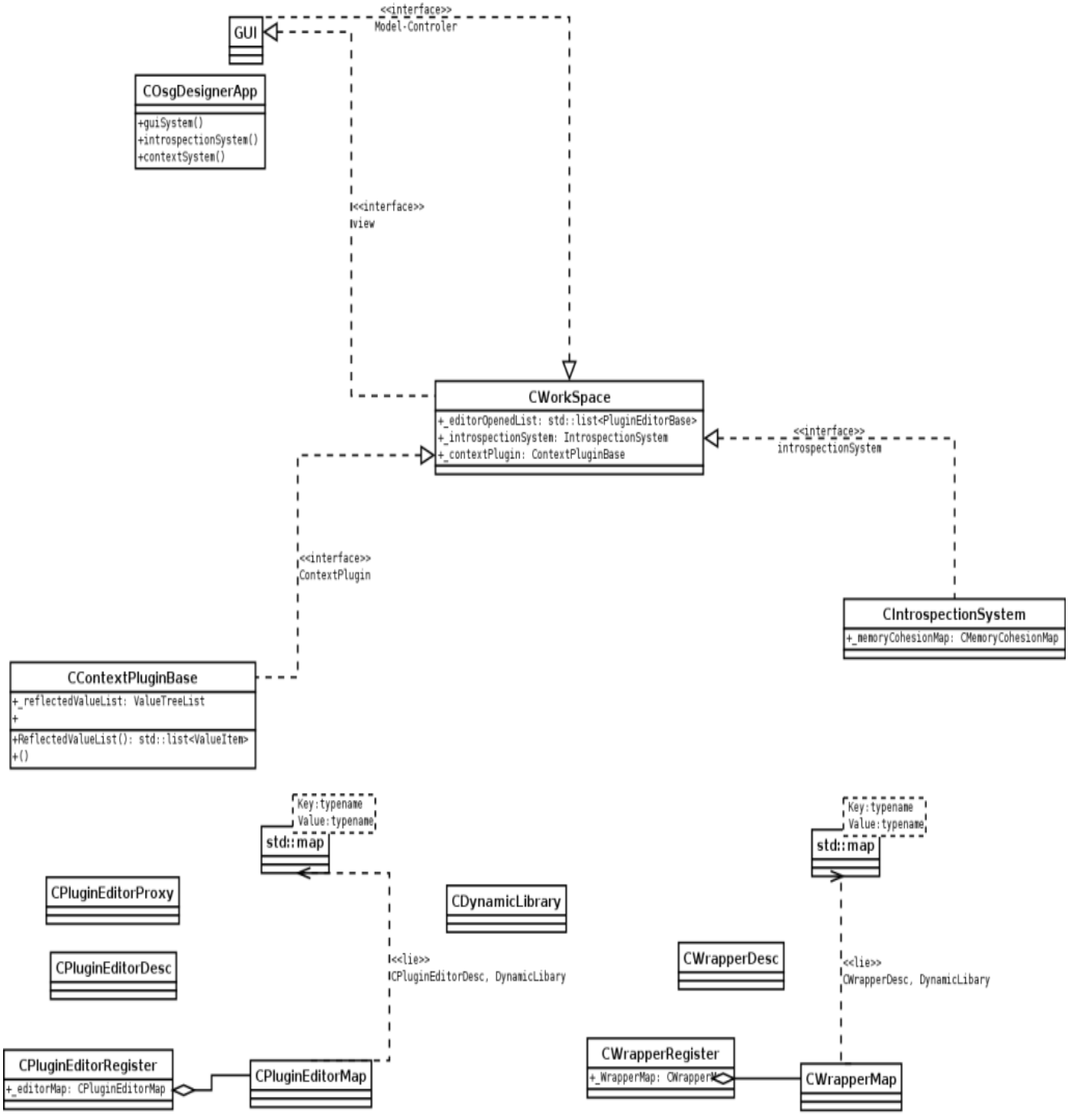


pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

3.2 – Technical description

– **osgDesigner** :

we will divide the program into different parts (classes) to represent global work :
here comes the class diagram :



pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

In additon, in the GUI we have defined others parts wich will require work :

- CDynamicLibrary Manager :
  osgWrapper Load/Unload :
  display the available osgwrapper list and show it they are already loaded.
- CPluginManager :
  Plugin Load/Unload :
  display the available plugin list and show it they are already loaded.
- CTypeViewer.
  list the reflected types, and display their information.
- CValueViewer
  reflects values in a tree form, i'ts a base widget to be used with the 'plugin editor'.
- CContextPlugin
  add possibilities to the menu File->export/import and all this functionnalities.
  add a value list to be reflected with constrains.
  can add 'plugin editor's.
  can render the 3D scene.
  can add any functionnality to the context.

– **Map Converter** :

It is a tool for topographic data conversion. It is based on a plug-in system allowing to convert data from any format to the engine's file format. It selects which plug in to use by checking the extension of the input file.

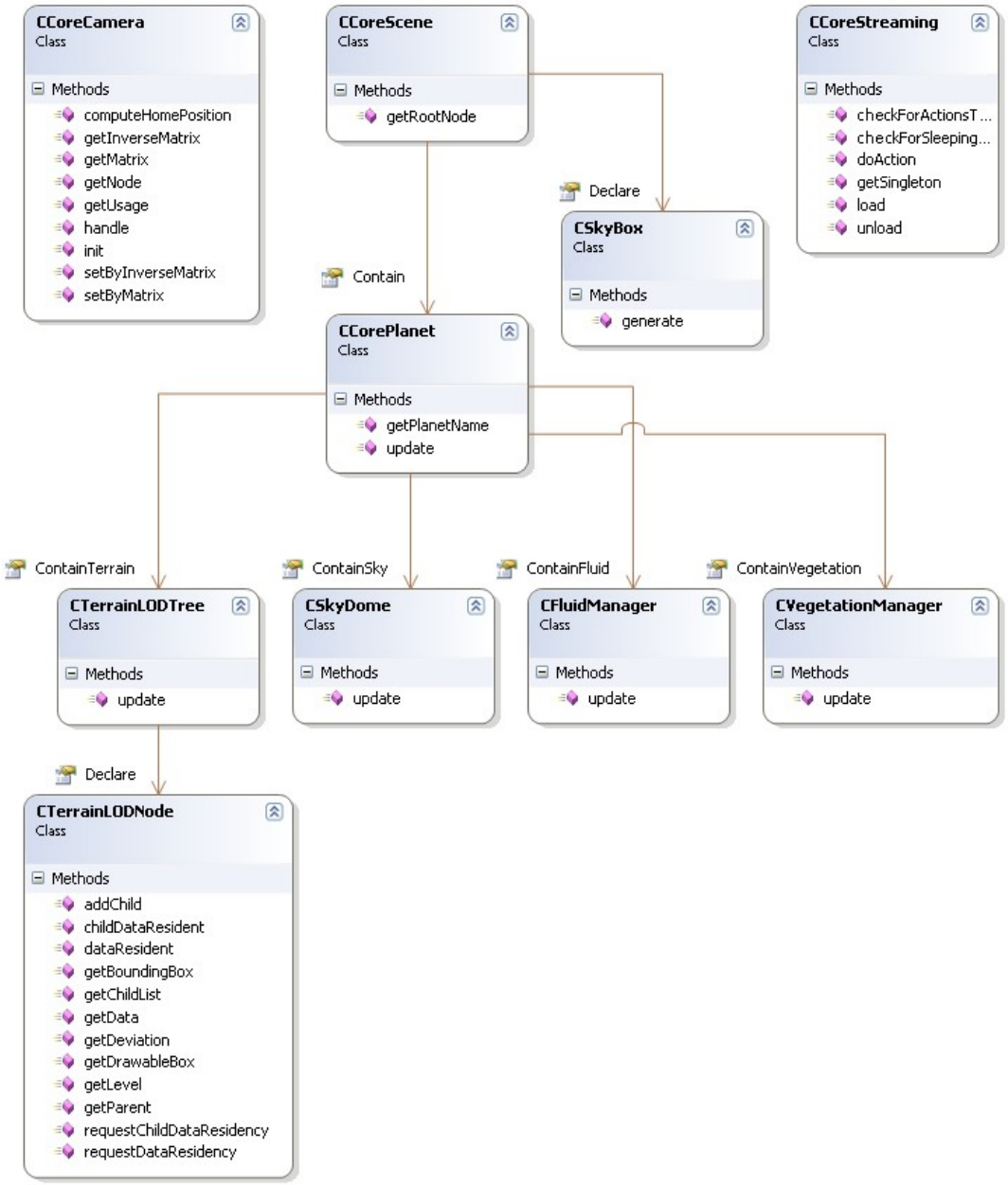
– **Magrathea** :

The engine is composed of many parts :

- mgtCore : it is the main part of the engine, it allows for example to manage the camera, the streaming system or the scene generated.
- mgtTerrain : it manages terrain's render and generation algorithm.
- mgtSky : this part allows to manage atmospheres and weather effects. It also contains Skydome and Skybox classes.
- mgtFluid : everything about water and fluids rendering is contained in this part.
- mgtVegetation : it allows to control vegetation rendering on the generated terrain.

Here is the class diagram representing this organization :

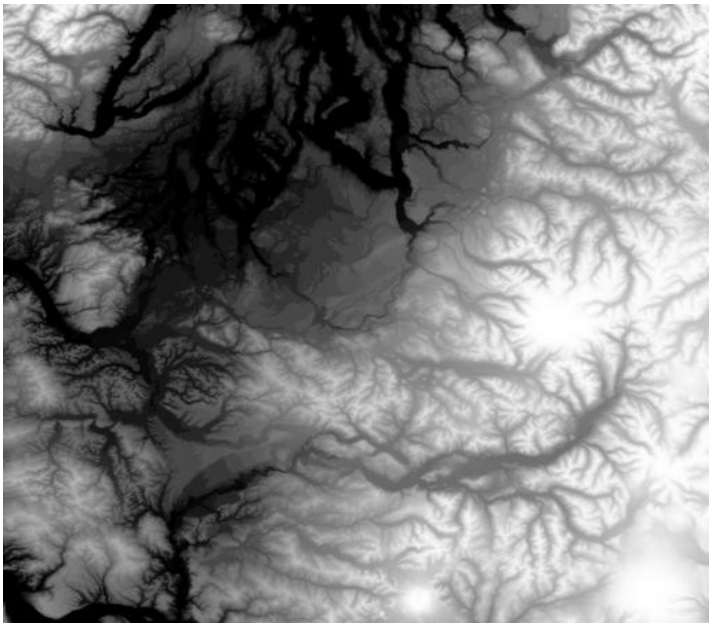
pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU



pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

3.3 – In/Out data

- **osgDesigner** :
  - Input : Xml configuration file, to remember user references for the graphical user interface and the plugins to load.
  - Output : Xml Configuration file, so the user can update and save his data. Possibility to export the 3D scene to other formats, the context plugin defines his own file format. Here the Magrathea context plugin will allow using the xml Magrathea file format.
- **Map Converter** :
  - Input : it will take in input topographic data, that gives informations about the elevation of the terrain. These files can be found at BT, DEM, or PNG file format.



Example of elevation file

  - Output : a MGT file, that follows a level of detail algorithm to store the transformed data. This way, the data is ready for rendering and the engine just have to load it.

- **Magrathea** :
  - Input : It takes as an input a XML file with configuration's options and that describe the solar system's organization. Here is the XSD description of this XML file :

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="Magrathea">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="GeneralConfiguration">
    </xs:element>
    <xs:element name="ChildList">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Planet">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="Configuration">
           <xs:complexType>
            <xs:attribute name="File" type="xs:string" use="required" />
           </xs:complexType>
          </xs:element>
          <xs:element name="ChildList" type="xs:string" />
         </xs:sequence>
         <xs:attribute name="Name" type="xs:string" use="required" />
         <xs:attribute name="Distance" type="xs:unsignedByte" use="required" />
         <xs:attribute name="RotationSpeed" type="xs:unsignedByte"
       use="required" />
         <xs:attribute name="RotationAroundItselfSpeed" type="xs:unsignedByte"
                  use="required" />
         <xs:attribute name="RotationAngle" type="xsunsignedByte" use="required"
                                    />
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Each planet file has a XML configuration file defined as follow :

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="MagratheaPlanet">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Terrain">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="File">
        <xs:complexType>
         <xs:attribute name="Path" type="xs:string" use="required" />
        </xs:complexType>
       </xs:element>
       <xs:element name="Texture">
        <xs:complexType>
         <xs:attribute name="Path" type="xs:string" use="required" />
        </xs:complexType>
       </xs:element>
       <xs:element name="Size">
        <xs:complexType>
         <xs:attribute name="Radius" type="xs:unsignedShort" use="required" />
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
```


pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

```
</xs:schema>
```

- Output : The engine's output is graphical, it is the generated environment. No file is created.

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

4. Annex

4.1 – General code's norm

- Class name : C(NameOfThePart)(NameOfTheClass)
  Examples :
  CCoreScene
  CSkyBox

- Class header's file name : (NameOfTheClass).h
  Examples :
  CCorePlanet.h
  CTerrainManager.h

- Class code's file name : (NameOfTheClass).cpp
  Examples :
  CCorePlanet.cpp
  CTerrainManager.cpp

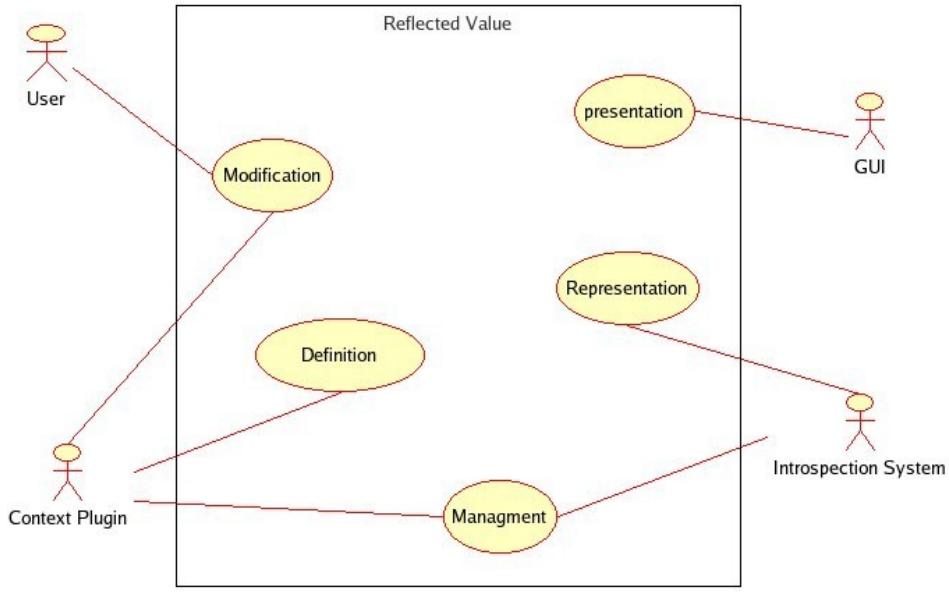
- Class' methods name : myMethodName()
  Examples :
  getChildList()
  getBoundingBox()

Example of class definition file :

```
#ifndef		__CCORETEST_H__
#define		__CCORETEST_H__

#include	"mgtCore/CoreExport.h"

namespace mgt
{
	class	MGT_EXPORT		CCoreTest
	{
	private:

	public:

		CCoreTest();
		~CCoreTest();

		void		test(void);
	};
}

#endif
```


pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

4.2 – osgDesigner Use Case



4.3 – osgDesigner code norm

- A file represents a system component and not a class.
- Header filies :
  indentation = 4 spaces.
  The code is between this #define
  ```
  #ifndef __HEADERFILENAME_H__
  #define __HEADERFILENAME_H__ 1
  #endif // ** __HEADERFILENAME_H__ ** //
  ```
- The class name begins with :
  - I = Interface.
  - T = Template.
  - S = Singleton.
  - C = others.

- Public methods :
  - constructor/destructor
  - accessors (const)
  - modificators (others)

- Public properties.
- Protected.
- Private.
- All inlines methods are defined after the class declaration.

pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU

Example :

```
class classe1;
class classe2;

namespace nomdunamespace
{
   class CnomDeLaClasse
   {
      public:

         // ** constructor / destructor ** //
         CNomDeLaClasse();
         ~CNomDeLaClasse();

         // ** accessor ** //
         float             getFloat() const;
         bool              isValid() const;
         bool              havChild() const;

         // ** modificator ** //
         void                 setFloat(float const val);
         void                 setClasse1(Classe1 const& classe);


      protected:

      private:

         static int        s_valInt;
         float             _valFloat;
         classe1*          _valInt;
   };

 std::ostream& operator <<(std::ostream& output, CNomDeLaClasse classe)
 {
  while
  return (output);
 };

}
```


pfe@epitech.net
Director : Flavien ASTRAUD
Technical responsible : Jérôme LANDRIEU